AppSec in a DevOps World

Peter Chestna, Director of Developer Engagement
Who am I?

• 25 Years Software Development Experience
• 10+ Years Application Security Experience
• Certified Agile Product Owner and Scrum Master
• At Veracode since 2006
  • From Waterfall to Agile to DevOps
  • From Monolith to MicroService
  • Consultant on DevSecOps best practices
• Fun Fact: I love whiskey!

@PeteChestna
Goals

- Why is AppSec important?
- How is DevOps changing application development?
- How is AppSec traditionally done?
- What needs to change?
  - What to build
  - What to measure
  - How to help
Applications are as risky as ever

- 35% of all applications used some kind of hard-coded password
- 39% of all applications use broken or risky cryptographic algorithms
- 28% of all applications were vulnerable to open redirect attacks
- 16% of all applications mix trusted and untrusted data in the same data structure or message
Majority of internally developed applications fail OWASP
U.S. Department of Homeland Security (DHS) research found that **90 percent** of security incidents result from exploits against defects in software.

“Is poor software development the biggest cyber threat?” CSOonline.com, September 2, 2015.
High Profile Breaches
All attacked through the app layer

TARGET

HOW: Sophisticated kill chain including exploitation of a vulnerable web application

RESULT: Hackers stole names, mailing addresses, phone numbers and email addresses from over 70 million shoppers

JPMORGAN CHASE

HOW: Vulnerability on website built and maintained by a third-party vendor in support of a charity

RESULT: Usernames and passwords for 76 million households and 7 million businesses accounts were stolen

COMMUNITY HEALTH

HOW: Targeted a flaw in OpenSSL, CVE-2014-0160, better known as Heartbleed

RESULT: The theft of Social Security numbers and other personal data belonging to 4.5 million patients
Business Mandate

© 2016 VERACODE INC.
Compressed Timelines

Waterfall

1-4 Releases Per Year

Agile

12-24 Releases Per Year

DevOps

100+ Releases Per Year
“DevOps is a cultural and professional movement, focused on how we build and operate high velocity organizations, born from the experiences of its practitioners.”

- Nathan Harvey (Chef)
Basic development cycle

1. Requirements
2. Analysis
3. Design
4. Coding
5. Testing
6. Acceptance

© 2016 VERACODE INC.
Not so different after all

Waterfall
- Requirements
- Analysis
- Design
- Coding
- Testing
- Acceptance

Agile
- Requirements
- Analysis
- Design
- Coding
- Testing
- Acceptance

DevOps

At Scale

Time
The diagram compares Waterfall, Agile, and DevOps methodologies in terms of Business Intent, Application Knowledge, and Operations Knowledge.

- **Waterfall**:
  - Business Intent
  - Application Knowledge
  - Operations Knowledge

- **Agile**:
  - Business Intent
  - Application Knowledge
  - Operations Knowledge

- **DevOps**:
  - Business Intent
  - Application Knowledge
  - Operations Knowledge

The diagram highlights continuity through DevOps, where knowledge is maintained and handoffs are reduced compared to the Waterfall and Agile approaches.
Agile - Process

5. Stand-up
6. Track Visibly

Z.P.

1. Backlog

Budget

Track Electronically

10. Review & Reflect

Really finish work

K10g
Transformation - Technology

Waterfall

Agile

DevOps
Is this your current AppSec program?
They/We know it’s coming…
Which outcome do you see?
DevOps – Process: Where is security?
Strategy

- Integration & Automation
- 3-legged barstool:
  - Training
  - Remediation Coaching
  - Scan early & often
Strategy – Integration & Automation

1. Develop
2. Build & Test
3. Static Analysis
4. Check in
5. Build
6. Static Analysis
   - Unit Tests
7. Deploy to QA/Stage
8. Dynamic Analysis
   - Regression Testing
9. Static Analysis

Flow:
- Backlog → Develop → Check in
- Develop → Build & Test → Static Analysis
- Build & Test → Check in → CI/CD Pipeline
- CI/CD Pipeline → Build
- Build → Static Analysis → Unit Tests
- Static Analysis → Unit Tests → Pass?
- Pass? → Yes → Deploy to QA/Stage
- Deploy to QA/Stage → Dynamic Analysis
- Dynamic Analysis → Regression Testing
- Regression Testing → Pass?
- Pass? → Yes → Stage then Prod
Strategy - Training

- Security teams can help developers by providing training, either through eLearning or in-person Instructor Led Training
- Think about targeted training based on policy violations

*State of Software Security Report: Focus on Industry Verticals, Volume 6, Veracode*
Get smart on DevOps

Train beyond your walls
For applications that used remediation coaching, development teams fixed more than 2.5x the average # of flaws per megabyte.
Applications that used sandbox had an average fix rate of 59%, or a 2x improvement in fix rate.

Developer Sandboxes help scanning branches and fixing security flaws before merging into Master or Release branches to avoid alerting the security team.

Policy Scans assess for official compliance & report into GRC systems.
DevOps – Pervasive Security

Training
(eLearning, instructor led, metadata driven)

Static Application Security Testing + 3rd Party Risk Analysis

Dynamic Application Security Testing

Runtime Application Self Protection

Threat Modeling
Security Grooming
Secure Design

Remediation and Mitigation Guidance
Secure Code Reviews

Manual Penetration Testing
Red Team Activities
Thank You!